

Android 系统的超级终端命令大全

一、	BusyBox 工具集	1
二、	安装和登录命令	2
	reboot	2
	mount	2
	umount	4
	exit	4
三、	文件处理命令	4
	mkdir	4
	grep	5
	dd	5
	find	6
	mv	7
	ls	8
	diff	9
	cmp	9
	cat	9
	ln	10
四、	系统管理命令	11
	df	11
	top	11
	free	12
	kill	13
五、	网络操作命令	14
	ifconfig	14
	ip	15
	ping	16
	netstat	16
	telnet	18
	route	18
六、	Android 系统的超级终端命令大全（六）--系统安全相关命令	19
	su	19
	umask	20
	chgrp	20
	chmod	21
	chown	22
	chattr	23

ps.....	24
七、 其他命令.....	24
tar	24
unzip	26
gunzip	27

一、 BusyBox 工具集

Android 系统所有命令都是运用超级终端下的 BusyBox 工具集。以下简称 BusyBox:

BusyBox 是标准 Linux 工具的一个单个可执行实现。BusyBox 包含了一些简单的工具,例如 cat 和 echo,还包含了一些更大、更复杂的工具,例如 grep、find、mount 以及 telnet。有些人将 BusyBox 称为 Linux 工具里的瑞士军刀。简单的说 BusyBox 就好像是个大工具箱,它集成压缩了 Linux 的许多工具和命令。

1、BusyBox 的诞生

BusyBox 最初是由 Bruce Perens 在 1996 年为 Debian GNU/Linux 安装盘编写的。其目标是在一张软盘上创建一个可引导的 GNU/Linux 系统,这可以用作安装盘和急救盘。

2、busybox 的用法

可以这样用 busybox

```
#busybox ls
```

他的功能就相当运行 ls 命令。

最常用的用法是建立指向 busybox 的链接,不同的链接名完成不同的功能。

```
#ln -s busybox ls
#ln -s busybox rm
#ln -s busybox mkdir
```

然后分别运行这三个链接:

```
#./ls
#./rm
#./mkdir
```

就可以分别完成了 ls rm 和 mkdir 命令的功能.虽然他们都指向同一个可执行程序 busybox,但是只要链接名不同,完成的功能就不同,很多 linux 网站都提供 busybox 的源代码下载。

3、配置 busybox

busybox 的配置程序和 linux 内核菜单配置方式简直一模一样。熟悉用 make menuconfig 方式配置 linux 内核的朋友很容易上手。

```
#cp busybox-1.00.tar.gz /babylinux
#cd /babylinux
#tar xvfz busybox-1.00.tar.gz
#cd busybox-1.00
#make menuconfig
```

下面是需要编译进 busybox 的功能选项。

General Configuration 应该选的选项:

```
Show verbose applet usage messages
Runtime SUID/SGID configuration via /etc/busybox.conf
Build Options
Build BusyBox as a static binary (no shared libs)
```

这个选项是一定要选择的，这样才能把 busybox 编译成静态链接的可执行文件，运行时才独立于其他函数库。否则必需要其他库文件才能运行，在单个 linux 内核不能使它正常工作。

Installation Options
Don't use /usr

这个选项也一定要选，否则 make install 后 busybox 将安装在原系统的/usr 下，这将覆盖掉系统原有的命令。选择这个选项后，make install 后会在 busybox 目录下生成一个叫_install 的目录，里面有 busybox 和指向它的链接。

其它选项都是一些 linux 基本命令选项，自己需要哪些命令就编译进去，一般用默认的就可以了，配置好后退出并保存。

4、编译并安装 busybox

```
#make  
#make install
```

编译好后在 busybox 目录下生成子目录_install，里面的内容：

```
drwxr-xr-x 2 root root 4096 11 月 24 15:28 bin  
rwxrwxrwx 1 root root 11 11 月 24 15:28 linuxrc -> bin/busybox  
drwxr-xr-x 2 root root 4096 11 月 24 15:28 sbin
```

其中可执行文件 busybox 在 bin 目录下，其他的都是指向他的符号链接。

二、 安装和登录命令

reboot

1. 作用

reboot 命令的作用是重新启动计算机，它的使用权限是系统管理者。

2. 格式

```
reboot [-n] [-w] [-d] [-f] [-i]
```

3. 主要参数

- n: 在重开机前不做将记忆体资料写回硬盘的动作。
- w: 并不会真的重开机，只是把记录写到/var/log/wtmp 文件里。
- d: 不把记录写到/var/log/wtmp 文件里（-n 这个参数包含了一d）。
- i: 在重开机之前先把所有与网络相关的装置停止。

mount

1. 作用

mount 命令的作用是加载文件系统，它的用权限是超级用户或/etc/fstab 中允许的使用者。

2. 格式

```
mount -a [-fv] [-t vfstype] [-n] [-rw] [-F] device dir
```

3. 主要参数

- h: 显示辅助信息。
- v: 显示信息，通常和-f 用来除错。
- a: 将/etc/fstab 中定义的所有文件系统挂上。

-F: 这个命令通常和-a 一起使用, 它会为每一个 mount 的动作产生一个行程负责执行。在系统需要挂上大量 NFS 文件系统时可以加快加载的速度。

-f: 通常用于除错。它会使 mount 不执行实际挂上的动作, 而是模拟整个挂上的过程, 通常会和-v 一起使用。

-t vfstype: 显示被加载文件系统的类型。

-n: 一般而言, mount 挂上后会在/etc/mntab 中写入一笔资料, 在系统中没有可写入文件系统的情况下, 可以用这个选项取消这个动作。

4. 应用技巧

在 Linux 和 Unix 系统上, 所有文件都是作为一个大型树 (以/为根) 的一部分访问的。要访问 CD-ROM 上的文件, 需要将 CD-ROM 设备挂装在文件树中的某个挂装点。如果发行版安装了自动挂装包, 那么这个步骤可自动进行。在 Linux 中, 如果要使用硬盘、光驱等储存设备, 就得先将它加载, 当储存设备挂上了之后, 就可以把它当成一个目录来访问。挂上一个设备使用 mount 命令。在使用 mount 这个指令时, 至少要先知道下列三种信息: 要加载对象的文件系统类型、要加载对象的设备名称及要将设备加载到哪个目录下。

(1) Linux 可以识别的文件系统

- ◆ Windows 95/98 常用的 FAT 32 文件系统: vfat ;
- ◆ Win NT/2000 的文件系统: ntfs ;
- ◆ OS/2 用的文件系统: hpfs;
- ◆ Linux 用的文件系统: ext2、ext3;
- ◆ CD-ROM 光盘用的文件系统: iso9660。

虽然 vfat 是指 FAT 32 系统, 但事实上它也兼容 FAT 16 的文件系统类型。

(2) 确定设备的名称

在 Linux 中, 设备名称通常都存在/dev 里。这些设备名称的命名都是有规则的, 可以用“推理”的方式把设备名称找出来。例如, /dev/hda1 这个 IDE 设备, hd 是 Hard Disk (硬盘) 的, sd 是 SCSI Device, fd 是 Floppy Device (或是 Floppy Disk?)。a 代表第一个设备, 通常 IDE 接口可以接上 4 个 IDE 设备 (比如 4 块硬盘)。所以要识别 IDE 硬盘的方法分别就是 hda、hdb、hdc、hdd。hda1 中的“1”代表 hda 的第一个硬盘分区 (partition), hda2 代表 hda 的第二主分区, 第一个逻辑分区从 hda5 开始, 依此类推。此外, 可以直接检查/var/log/messages 文件, 在该文件中可以找到计算机开机后系统已辨认出来的设备代号。

(3) 查找挂接点

在决定将设备挂接之前, 先要查看一下计算机是不是有个/mnt 的空目录, 该目录就是专门用来当作挂载点 (MountPoint) 的目录。建议在/mnt 里建几个 /mnt/cdrom、/mnt/floppy、/mnt/mo 等目录, 当作目录的专用挂载点。举例而言, 如要挂载下列 5 个设备, 其执行指令可能如下 (假设都是 Linux 的 ext2 系统, 如果是 Windows XX 请将 ext2 改成 vfat):

```
软盘 ==>mount -t ext2 /dev/fd0 /mnt/floppy
```

```
cdrom ==>mount -t iso9660 /dev/hdc /mnt/cdrom
```

```
SCSI cdrom ==>mount -t iso9660 /dev/sdb /mnt/scdrom
```

```
SCSI cdr ==>mount -t iso9660 /dev/sdc /mnt/scdr
```

不过目前大多数较新的 Linux 发行版本 (包括红旗 Linux、中软 Linux、Mandrake Linux 等) 都可以自动挂装文件系统, 但 Red Hat Linux 除外。

umount

1. 作用

umount 命令的作用是卸载一个文件系统，它的使用权限是超级用户或 /etc/fstab 中允许的使用者。

2. 格式

umount -a [-fFnrsvw] [-t vfstype] [-n] [-rw] [-F] device dir

3. 使用说明

umount 命令是 mount 命令的逆操作，它的参数和使用方法和 mount 命令是一样的。Linux 挂装 CD-ROM 后，会锁定 CD-ROM，这样就不能用 CD-ROM 面板上的 Eject 按钮弹出它。但是，当不再需要光盘时，如果已将 /cdrom 作为符号链接，请使用 umount/cdrom 来卸装它。仅当无用户正在使用光盘时，该命令才会成功。该命令包括了将带有当前工作目录当作该光盘中的目录的终端窗口。

exit

1. 作用

exit 命令的作用是退出系统，它的使用权限是所有用户。

2. 格式

exit

3. 参数

exit 命令没有参数，运行后退出系统进入登录界面。

三、 文件处理命令

mkdir

1. 作用

mkdir 命令的作用是建立名称为 dirname 的子目录，与 MS DOS 下的 md 命令类似，它的使用权限是所有用户。

2. 格式

mkdir [options] 目录名

3. [options]主要参数

-m, --mode=模式：设定权限，与 chmod 类似。

-p, --parents：需要时创建上层目录；如果目录早已存在，则不当作错误。

-v, --verbose：每次创建新目录都显示信息。

--version：显示版本信息后离开。

4. 应用实例

在进行目录创建时可以设置目录的权限，此时使用的参数是“-m”。假设要创建的目录名是“tsk”，让所有用户都有 rwx（即读、写、执行的权限），那么可以使用以下命令：

\$ mkdir -m 777 tsk

grep

1. 作用

grep 命令可以指定文件中搜索特定的内容，并将含有这些内容的行标准输出。**grep** 全称是 **Global Regular Expression Print**，表示全局正则表达式版本，它的使用权限是所有用户。

2. 格式

grep [options]

3. 主要参数

[options]主要参数：

- c: 只输出匹配行的计数。
- I: 不区分大小写（只适用于单字符）。
- h: 查询多文件时不显示文件名。
- l: 查询多文件时只输出包含匹配字符的文件名。
- n: 显示匹配行及行号。
- s: 不显示不存在或无匹配文本的错误信息。
- v: 显示不包含匹配文本的所有行。

pattern 正则表达式主要参数：

- \: 忽略正则表达式中特殊字符的原有含义。
- ^: 匹配正则表达式的开始行。
- \$: 匹配正则表达式的结束行。
- \: 到匹配正则表达式的行结束。
- []: 单个字符，如[A]即 A 符合要求。
- [-]: 范围，如[A-Z]，即 A、B、C 一直到 Z 都符合要求。
- .: 所有的单个字符。
- *: 有字符，长度可以为 0。

正则表达式是 **Linux/Unix** 系统中非常重要的概念。正则表达式（也称为“**regex**”或“**regexp**”）是一个可以描述一类字符串的模式（**Pattern**）。如果一个字符串可以用某个正则表达式来描述，我们就说这个字符和该正则表达式匹配（**Match**）。这和 **DOS** 中用户可以使用通配符“*”代表任意字符类似。在 **Linux** 系统上，正则表达式通常被用来查找文本的模式，以及对文本执行“搜索—替换”操作和其它功能。

4. 应用实例

查询 **DNS** 服务是日常工作之一，这意味着要维护覆盖不同网络的大量 **IP** 地址。有时 **IP** 地址会超过 2000 个。如果要查看 **nnn.nnn** 网络地址，但是却忘了第二部分中的其余部分，只知到有两个句点，例如 **nnn nn..**。要抽取其中所有 **nnn.nnn** **IP** 地址，使用 `[0-9]\{3\}\.[0-0]\{3\}`。含义是任意数字出现 3 次，后跟句点，接着是任意数字出现 3 次，后跟句点。`$grep '[0-9]\{3\}\.[0-0]\{3\}' ipfile`

补充说明，**grep** 家族还包括 **fgrep** 和 **egrep**。**fgrep** 是 **fix grep**，允许查找字符串而不是一个模式；**egrep** 是扩展 **grep**，支持基本及扩展的正则表达式，但不支持 **\q** 模式范围的应用及与之相对应的一些更加规范的模式。

dd

1. 作用

dd 命令用来复制文件，并根据参数将数据转换和格式化。

2. 格式

dd [options]

3. [options]主要参数

bs=字节：强迫 ibs=及 obs=。

cbs=字节：每次转换指定的。

conv=关键字：根据以逗号分隔的关键字表示的方式来转换文件。

count=块数目：只复制指定的输入数据。

ibs=字节：每次读取指定的。

if=文件：读取内容，而非标准输入的数据。

obs=字节：每次写入指定的。

of=文件：将数据写入，而不在标准输出显示。

seek=块数目：先略过以 obs 为单位的指定的输出数据。

skip=块数目：先略过以 ibs 为单位的指定的输入数据。

4. 应用实例

dd 命令常常用来制作 Linux 启动盘。先找一个可引导内核，令它的根设备指向正确的根分区，然后使用 dd 命令将其写入软盘：

```
$ rdev vmlinuz /dev/had
```

```
$dd if=vmlinuz of=/dev/fd0
```

上面代码说明，使用 rdev 命令将可引导内核 vmlinuz 中的根设备指向 /dev/hda，请把“had”换成自己的根分区，接下来用 dd 命令将该内核写入软盘。

find

1. 作用

find 命令的作用是在目录中搜索文件，它的使用权限是所有用户。

2. 格式

find [path][options][expression]

path 指定目录路径，系统从这里开始沿着目录树向下查找文件。它是一个路径列表，相互用空格分离，如果不写 path，那么默认为当前目录。

3. 主要参数

[options]参数：

-depth：使用深度级别的查找过程方式，在某层指定目录中优先查找文件内容。

-maxdepth levels：表示至多查找到开始目录的第 level 层子目录。level 是一个非负数，如果 level 是 0 的话表示仅在当前目录中查找。

-mindepth levels：表示至少查找到开始目录的第 level 层子目录。

-mount：不在其它文件系统（如 Msdos、Vfat 等）的目录和文件中查找。

-version：打印版本。

[expression]是匹配表达式，是 find 命令接受的表达式，find 命令的所有操作都是针对表达式的。它的参数非常多，这里只介绍一些常用的参数。

-name：支持通配符*和?。

-atime n：搜索在过去 n 天读取过的文件。

-ctime n：搜索在过去 n 天修改过的文件。

-group grpoupname: 搜索所有组为 grpoupname 的文件。
-user 用户名: 搜索所有文件属主为用户名 (ID 或名称) 的文件。
-size n: 搜索文件大小是 n 个 block 的文件。
-print: 输出搜索结果, 并且打印。

4. 应用技巧

find 命令查找文件的几种方法:

(1) 根据文件名查找

例如, 我们想要查找一个文件名是 lilo.conf 的文件, 可以使用如下命令:

```
find / -name lilo.conf
```

find 命令后的 “/” 表示搜索整个硬盘。

(2) 快速查找文件

根据文件名查找文件会遇到一个实际问题, 就是要花费相当长的一段时间, 特别是大型 Linux 文件系统和大容量硬盘文件放在很深的子目录中时。如果我们知道了这个文件存放在某个目录中, 那么只要在这个目录中往下寻找就能节省很多时间。比如 smb.conf 文件, 从它的文件后缀 “.conf” 可以判断这是一个配置文件, 那么它应该在 /etc 目录内, 此时可以使用下面命令:

```
find /etc -name smb.conf
```

这样, 使用 “快速查找文件” 方式可以缩短时间。

(3) 根据部分文件名查找方法

有时我们知道只某个文件包含有 abvd 这 4 个字, 那么要查找系统中所有包含有这 4 个字符的文件可以输入下面命令:

```
find / -name '*abvd*'
```

输入这个命令以后, Linux 系统会将在 / 目录中查找所有的包含有 abvd 这 4 个字符的文件 (其中 * 是通配符), 比如 abvdmyz 等符合条件的文件都能显示出来。

(4) 使用混合查找方式查找文件

find 命令可以使用混合查找的方法, 例如, 我们想在 /etc 目录中查找大于 500000 字节, 并且在 24 小时内修改的某个文件, 则可以使用 -and (与) 把两个查找参数链接起来组合成一个混合的查找方式。

```
find /etc -size +500000c -and -mtime +1
```

mv

1. 作用

mv 命令用来为文件或目录改名, 或者将文件由一个目录移入另一个目录中, 它的使用权限是所有用户。该命令如同 DOS 命令中的 ren 和 move 的组合。

2. 格式

mv[options] 源文件或目录 目标文件或目录

3. [options] 主要参数

-i: 交互方式操作。如果 mv 操作将导致对已存在的目标文件的覆盖, 此时系统询问是否重写, 要求用户回答 “y” 或 “n”, 这样可以避免误覆盖文件。

-f: 禁止交互操作。mv 操作要覆盖某个已有的目标文件时不给任何指示, 指定此参数后 i 参数将不再起作用。

4. 应用实例

(1) 将 /usr/cbu 中的所有文件移到当前目录 (用 “.” 表示) 中:

```
$ mv /usr/cbu/ * .  
(2) 将文件 cjh.txt 重命名为 wjz.txt:  
$ mv cjh.txt wjz.txt
```

ls

1. 作用

ls 命令用于显示目录内容，类似 DOS 下的 dir 命令，它的使用权限是所有用户。

2. 格式

ls [options][filename]

3. [options]主要参数

- a, --all: 不隐藏任何以 “.” 字符开始的项目。
- A, --almost-all: 列出除了 “.” 及 “..” 以外的任何项目。
- author: 印出每个文件著作者。
- b, --escape: 以八进制溢出序列表示不可打印的字符。
- block-size=大小: 块以指定的字节为单位。
- B, --ignore-backups: 不列出任何以 ~ 字符结束的项目。
- f: 不进行排序，-aU 参数生效，-lst 参数失效。
- F, --classify: 加上文件类型的指示符号(*/=@| 其中一个)。
- g: like -l, but do not list owner。
- G, --no-group: inhibit display of group information。
- i, --inode: 列出每个文件的 inode 号。
- I, --ignore=样式: 不印出任何符合 Shell 万用字符的项目。
- k: 即-block-size=1K。
- l: 使用较长格式列出信息。
- L, --dereference: 当显示符号链接的文件信息时，显示符号链接所指示的对象，而并非符号链接本身的信息。
- m: 所有项目以逗号分隔，并填满整行行宽。
- n, --numeric-uid-gid: 类似-l，但列出 UID 及 GID 号。
- N, --literal: 列出未经处理的项目名称，例如不特别处理控制字符。
- p, --file-type: 加上文件类型的指示符号 (/=@| 其中一个)。
- Q, --quote-name: 将项目名称括上双引号。
- r, --reverse: 依相反次序排列。
- R, --recursive: 同时列出所有子目录层。
- s, --size: 以块大小为序。

4. 应用举例

ls 命令是 Linux 系统使用频率最多的命令，它的参数也是 Linux 命令中最多的。使用 ls 命令时会有几种不同的颜色，其中蓝色表示是目录，绿色表示是可执行文件，红色表示是压缩文件，浅蓝色表示是链接文件，加粗的黑色表示符号链接，灰色表示是其它格式文件。ls 最常使用的是 ls-l。

文件类型开头是由 10 个字符构成的字符串。其中第一个字符表示文件类型，它可以是下述类型之一：-（普通文件）、d（目录）、l（符号链接）、b（块设备文件）、c（字符设备文件）。后面的 9 个字符表示文件的访问权限，分为 3

组，每组 3 位。第一组表示文件属主的权限，第二组表示同组用户的权限，第三组表示其他用户的权限。每一组的三个字符分别表示对文件的读（r）、写（w）和执行权限（x）。对于目录，表示进入权限。s 表示当文件被执行时，把该文件的 UID 或 GID 赋予执行进程的 UID（用户 ID）或 GID（组 ID）。t 表示设置标志位（留在内存，不被换出）。如果该文件是目录，那么在该目录中的文件只能被超级用户、目录拥有者或文件属主删除。如果它是可执行文件，那么在该文件执行后，指向其正文段的指针仍留在内存。这样再次执行它时，系统就能更快地装入该文件。接着显示的是文件大小、生成时间、文件或命令名称。

diff

1. 作用

diff 命令用于两个文件之间的比较，并指出两者的不同，它的使用权限是所有用户。

2. 格式

diff [options] 源文件 目标文件

3. [options]主要参数

- a: 将所有文件当作文本文件来处理。
- b: 忽略空格造成的不同。
- B: 忽略空行造成的不同。
- c: 使用纲要输出格式。
- H: 利用试探法加速对大文件的搜索。
- I: 忽略大小写的变化。
- n --rcs: 输出 RCS 格式。

cmp

1. 作用

cmp（“compare”的缩写）命令用来简要指出两个文件是否存在差异，它的使用权限是所有用户。

2. 格式

cmp[options] 文件名

3. [options]主要参数

-l: 将字节以十进制的方式输出，并方便将两个文件中不同的以八进制的方式输出。

cat

1. 作用

cat（“concatenate”的缩写）命令用于连接并显示指定的一个和多个文件的有关信息，它的使用权限是所有用户。

2. 格式

cat [options] 文件 1 文件 2.....

3. [options]主要参数

- n: 由第一行开始对所有输出的行数编号。
- b: 和-n 相似, 只不过对于空白行不编号。
- s: 当遇到有连续两行以上的空白行时, 就代换为一行的空白行。

4. 应用举例

(1) `cat` 命令一个最简单的用处是显示文本文件的内容。例如, 我们想在命令行看一下 `README` 文件的内容, 可以使用命令:

```
$ cat README
```

(2) 有时需要将几个文件处理成一个文件, 并将这种处理的结果保存到一个单独的输出文件。`cat` 命令在其输入上接受一个或多个文件, 并将它们作为一个单独的文件打印到它的输出。例如, 把 `README` 和 `INSTALL` 的文件内容加上行号 (空白行不加) 之后, 将内容附加到一个新文本文件 `File1` 中:

```
$ cat README INSTALL File1
```

(3) `cat` 还有一个重要的功能就是可以对行进行编号。这种功能对于程序文档的编制, 以及法律和科学文档的编制很方便, 打印在左边的行号使得参考文档的某一部分变得容易, 这些在编程、科学研究、业务报告甚至是立法工作中都是非常重要的。对行进行编号功能有 `-b` (只能对非空白行进行编号) 和 `-n` (可以对所有行进行编号) 两个参数:

```
$ cat -b /etc/named.conf
```

ln

1. 作用

`ln` 命令用来在文件之间创建链接, 它的使用权限是所有用户。

2. 格式

```
ln [options] 源文件 [链接名]
```

3. 参数

- f: 链结时先将源文件删除。
- d: 允许系统管理者硬链结自己的目录。
- s: 进行软链结(Symbolic Link)。
- b: 将在链结时会被覆盖或删除的文件进行备份。

链接有两种, 一种被称为硬链接 (Hard Link), 另一种被称为符号链接 (Symbolic Link)。默认情况下, `ln` 命令产生硬链接。硬链接指通过索引节点来进行的连接。在 `Linux` 的文件系统中, 保存在磁盘分区中的文件不管是什么类型都给它分配一个编号, 称为索引节点号(InodeIndex)。在 `Linux` 中, 多个文件名指向同一索引节点是存在的。一般这种连接就是硬连接。硬连接的作用是允许一个文件拥有多个有效路径名, 这样用户就可以建立硬连接到重要文件, 以防止“误删”的功能。其原因如上所述, 因为对应该目录的索引节点有一个以上的连接。只删除一个连接并不影响索引节点本身和其它的连接, 只有当最后一个连接被删除后, 文件的数据块及目录的连接才会被释放。也就是说, 文件才会被真正删除。与硬连接相对应, `Linux` 系统中还存在另一种连接, 称为符号连接(Symbolic Link), 也叫软连接。软链接文件有点类似于 `Windows` 的快捷方式。它实际上是特殊文件的一种。在符号连接中, 文件实际上是一个文本文件, 其中包含的有另一文件的位置信息。

四、 系统管理命令

df

1. 作用
df 命令用来检查文件系统的磁盘空间占用情况，使用权限是所有用户。
2. 格式
df [options]
3. 主要参数
 - s: 对每个 Names 参数只给出占用的数据块总数。
 - a: 递归地显示指定目录中各文件及子目录中各文件占用的数据块数。若既不指定—s，也不指定—a，则只显示 Names 中的每一个目录及其中的各子目录所占的磁盘块数。
 - k: 以 1024 字节为单位列出磁盘空间使用情况。
 - x: 跳过在不同文件系统上的目录不予统计。
 - l: 计算所有的文件大小，对硬链接文件则计算多次。
 - i: 显示 inode 信息而非块使用量。
 - h: 以容易理解的格式印出文件系统大小，例如 136KB、254MB、21GB。
 - P: 使用 POSIX 输出格式。
 - T: 显示文件系统类型。
4. 说明
df 命令被广泛地用来生成文件系统的使用统计数据，它能显示系统中所有的文件系统的信息，包括总容量、可用的空闲空间、目前的安装点等。超级权限用户使用 df 命令时会发现这样的情况：某个分区的容量超过了 100%。这是因为 Linux 系统为超级用户保留了 10% 的空间，由其单独支配。也就是说，对于超级用户而言，他所见到的硬盘容量将是 110%。这样的安排对于系统管理而言是有好处的，当硬盘被使用的容量接近 100% 时系统管理员还可以正常工作。
5. 应用实例

Linux 支持的文件系统非常多，包括 JFS、ReiserFS、ext、ext2、ext3、ISO9660、XFS、Minx、vfat、MSDOS 等。使用 df -T 命令查看磁盘空间时还可以得到文件系统的信息：

```
# df -T
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/hda7	reiserfs	5.2G	1.6G	3.7G	30%	/
/dev/hda1	vfat	2.4G	1.6G	827M	66%	/windows/C
/dev/hda5	vfat	3.0G	1.7G	1.3G	57%	/windows/D
/dev/hda9	vfat	3.0G	2.4G	566M	82%	/windows/E
/dev/hda10	NTFS	3.2G	573M	2.6G	18%	/windows/F
/dev/hda11	vfat	1.6G	1.5G	23M	99%	/windows/G

从上面除了可以看到磁盘空间的容量、使用情况外，分区的文件系统类型、挂载点等信息也一览无遗。

top

1. 作用
top 命令用来显示执行中的程序进程，使用权限是所有用户。
2. 格式
top [-] [d delay] [q] [c] [S] [s] [n]
3. 主要参数
d: 指定更新的间隔，以秒计算。
q: 没有任何延迟的更新。如果使用者有超级用户，则 top 命令将会以最高的优先序执行。
c: 显示进程完整的路径与名称。
S: 累积模式，会将已完成或消失的子行程的 CPU 时间累积起来。
s: 安全模式。
i: 不显示任何闲置(Idle)或无用(Zombie)的行程。
n: 显示更新的次数，完成后将会退出 top。
4. 说明
top 命令是 Linux 系统管理的一个主要命令，通过它可以获得许多信息。
下面列出了详细解释。
PID (Process ID): 进程标示号。
USER: 进程所有者的用户名。
PR: 进程的优先级别。
NI: 进程的优先级别数值。
VIRT: 进程占用的虚拟内存值。
RES: 进程占用的物理内存值。
SHR: 进程使用的共享内存值。
S: 进程的状态，其中 S 表示休眠，R 表示正在运行，Z 表示僵死状态，N 表示该进程优先值是负数。
%CPU: 该进程占用的 CPU 使用率。
%MEM: 该进程占用的物理内存和总内存的百分比。
TIME+: 该进程启动后占用的总的 CPU 时间。
Command: 进程启动的启动命令名称，如果这一行显示不下，进程会有一个完整的命令行。
top 命令使用过程中，还可以使用一些交互的命令来完成其它参数的功能。这些命令是通过快捷键启动的。
P: 根据 CPU 使用大小进行排序。
T: 根据时间、累计时间排序。
q: 退出 top 命令。
m: 切换显示内存信息。
t: 切换显示进程和 CPU 状态信息。
c: 切换显示命令名称和完整命令行。
M: 根据使用内存大小进行排序。
W: 将当前设置写入 ~/.toprc 文件中。这是写 top 配置文件的推荐方法。
可以看到，top 命令是一个功能十分强大的监控系统的工具，对于系统管理员而言尤其重要。但是，它的缺点是会消耗很多系统资源。

1. 作用
free 命令用来显示内存的使用情况，使用权限是所有用户。
2. 格式
free [-b|-k|-m] [-o] [-s delay] [-t] [-V]
3. 主要参数
-b -k -m: 分别以字节（KB、MB）为单位显示内存使用情况。
-s delay: 显示每隔多少秒数来显示一次内存使用情况。
-t: 显示内存总和列。
-o: 不显示缓冲区调节列。
4. 应用实例
free 命令是用来查看内存使用情况的主要命令。和 top 命令相比，它的优点是使用简单，并且只占用很少的系统资源。通过 -S 参数可以使用 free 命令不间断地监视有多少内存存在使用，这样可以把它当作一个方便实时监控器。

```
# free -b -s5
```


使用这个命令后终端会连续不断地报告内存使用情况（以字节为单位），每 5 秒更新一次。

kill

1. 作用
kill 命令用来中止一个进程。
2. 格式
kill [-s signal | -p] [-a] pid ...
kill -l [signal]
3. 参数
-s: 指定发送的信号。
-p: 模拟发送信号。
-l: 指定信号的名称列表。
pid: 要中止进程的 ID 号。
Signal: 表示信号。
4. 说明
进程是 Linux 系统中一个非常重要的概念。Linux 是一个多任务的操作系统，系统上经常同时运行着多个进程。我们不关心这些进程究竟是如何分配的，或者是内核如何管理分配时间片的，所关心的是如何去控制这些进程，让它们能够很好地为用户服务。
Linux 操作系统包括三种不同类型的进程，每种进程都有自己的特点和属性。交互进程是由一个 Shell 启动的进程。交互进程既可以在前台运行，也可以在后台运行。
批处理进程和终端没有联系，是一个进程序列。监控进程（也称系统守护进程）是 Linux 系统启动时启动的进程，并在后台运行。例如，httpd 是著名的 Apache 服务器的监控进程。
kill 命令的工作原理是，向 Linux 系统的内核发送一个系统操作信号和某个程序的进程标识号，然后系统内核就可以对进程标识号指定的进程进行操作。比如在 top 命令中，我们看到系统运行许多进程，有时就需要使用 kill 中止某些进程

来提高系统资源。在讲解安装和登陆命令时，曾提到系统多个虚拟控制台的作用是当一个程序出错造成系统死锁时，可以切换到其它虚拟控制台工作关闭这个程序。此时使用的命令就是 `kill`，因为 `kill` 是大多数 Shell 内部命令可以直接调用的。

5.应用实例

(1) 强行中止（经常使用杀掉）一个进程标识号为 324 的进程：

```
#kill -9 324
```

(2) 解除 Linux 系统的死锁

在 Linux 中有时会发生这样一种情况：一个程序崩溃，并且处于死锁的状态。此时一般不用重新启动计算机，只需要中止(或者说是关闭)这个有问题的程序即可。当 `kill` 处于 X-Window 界面时，主要的程序(除了崩溃的程序之外)一般都已经正常启动了。此时打开一个终端，在那里中止有问题的程序。比如，如果 Mozilla 浏览器程序出现了锁死的情况，可以使用 `kill` 命令来中止所有包含有 Mozilla 浏览器的程序。首先用 `top` 命令查处该程序的 PID，然后使用 `kill` 命令停止这个程序：

```
#kill -SIGKILL XXX
```

其中，XXX 是包含有 Mozilla 浏览器的程序的进程标识号。

(3) 使用命令回收内存

我们知道内存对于系统是非常重要的，回收内存可以提高系统资源。`kill` 命令可以及时地中止一些“越轨”的程序或很长时间没有相应的程序。例如，使用 `top` 命令发现一个无用 (Zombie) 的进程，此时可以使用下面命令：

```
#kill -9 XXX
```

其中，XXX 是无用的进程标识号。

然后使用下面命令：

```
#free
```

此时会发现可用内存容量增加了。

(4) `killall` 命令

Linux 下还提供了一个 `killall` 命令，可以直接使用进程的名字而不是进程标识号，例如：

```
# killall -HUP inetd
```

五、 网络操作命令

ifconfig

1. 作用

`ifconfig` 用于查看和更改网络接口的地址和参数，包括 IP 地址、网络掩码、广播地址，使用权限是超级用户。

2. 格式

```
ifconfig -interface [options] address
```

3.主要参数

-interface: 指定的网络接口名，如 `eth0` 和 `eth1`。

up: 激活指定的网络接口卡。

down: 关闭指定的网络接口。

broadcast address: 设置接口的广播地址。

pointopoint: 启用点对点方式。
address: 设置指定接口设备的 IP 地址。
netmask address: 设置接口的子网掩码。

3. 应用说明

ifconfig 是用来设置和配置网卡的命令行工具。为了手工配置网络，这是一个必须掌握的命令。使用该命令的好处是无须重新启动机器。要赋给 **eth0** 接口 IP 地址 207.164.186.2，并且马上激活它，使用下面命令：

```
#ifconfig eth0 210.34.6.89 netmask 255.255.255.128 broadcast 210.34.6.127
```

该命令的作用是设置网卡 **eth0** 的 IP 地址、网络掩码和网络的本地广播地址。若运行不带任何参数的 **ifconfig** 命令，这个命令将显示机器所有激活接口的信息。带有“-a”参数的命令则显示所有接口的信息，包括没有激活的接口。注意，用 **ifconfig** 命令配置的网络设备参数，机器重新启动以后将会丢失。

如果要暂停某个网络接口的工作，可以使用 **down** 参数：

```
#ifconfig eth0 down
```

ip

1. 作用

ip 是 **iproute2** 软件包里面的一个强大的网络配置工具，它能够替代一些传统的网络管理工具，例如 **ifconfig**、**route** 等，使用权限为超级用户。几乎所有的 Linux 发行版本都支持该命令。

2. 格式

ip [OPTIONS] OBJECT [COMMAND [ARGUMENTS]]

3. 主要参数

OPTIONS 是修改 **ip** 行为或改变其输出的选项。所有的选项都是以-字符开头，分为长、短两种形式。

OBJECT 是要管理者获取信息的对象。

-V, -Version 打印 **ip** 的版本并退出。

-s, -stats, -statistics 输出更为详尽的信息。如果这个选项出现两次或多次，则输出的信息将更为详尽。

-f, -family 这个选项后面接协议种类，包括 **inet**、**inet6** 或 **link**，强调使用的协议种类。如果没有足够的信息告诉 **ip** 使用的协议种类，**ip** 就会使用默认值 **inet** 或 **any**。**link** 比较特殊，它表示不涉及任何网络协议。

-4 是 **-family inet** 的简写。

-6 是 **-family inet6** 的简写。

-0 是 **-family link** 的简写。

-o, -oneline 对每行记录都使用单行输出，回行用字符代替。如果需要使用 **wc**、**grep** 等工具处理 **ip** 的输出，则会用到这个选项。

-r, -resolve 查询域名解析系统，用获得的主机名代替主机 IP 地址

COMMAND

设置针对指定对象执行的操作，它和对象的类型有关。一般情况下，**ip** 支持对象的增加(**add**)、删除(**delete**)和展示(**show** 或 **list**)。有些对象不支持这些操作，或者有其它的一些命令。对于所有的对象，用户可以使用 **help** 命令获得帮助。这个命令会列出这个对象支持的命令和参数的语法。如果没有指定对象的操作命

令，ip 会使用默认的命令。一般情况下，默认命令是 list，如果对象不能列出，就会执行 help 命令。

ARGUMENTS

是命令的一些参数，它们依赖于对象和命令。ip 支持两种类型的参数：flag 和 parameter。flag 由一个关键词组成；parameter 由一个关键词加一个数值组成。为了方便，每个命令都有一个可以忽略的默认参数。例如，参数 dev 是 ip link 命令的默认参数，因此 ip link ls eth0 等于 ip link ls dev eth0。我们将在后面的详细介绍每个命令的使用，命令的默认参数将使用 default 标出。

4. 应用实例

添加 IP 地址 192.168.2.2/24 到 eth0 网卡上：

```
#ip addr add 192.168.1.1/24 dev eth0
```

丢弃源地址属于 192.168.2.0/24 网络的所有数据报：

```
#ip rule add from 192.168.2.0/24 prio 32777 reject
```

ping

1. 作用

ping 检测主机网络接口状态，使用权限是所有用户。

2. 格式

ping [-dfnqrRv][[-c][-i][-I][-l][-p][-s][-t] IP 地址

3.主要参数

- d: 使用 Socket 的 SO_DEBUG 功能。
- c: 设置完成要求回应的次数。
- f: 极限检测。
- i: 指定收发信息的间隔秒数。
- I: 网络界面使用指定的网络界面送出数据包。
- l: 前置载入，设置在送出要求信息之前，先行发出的数据包。
- n: 只输出数值。
- p: 设置填满数据包的范本样式。
- q: 不显示指令执行过程，开头和结尾的相关信息除外。
- r: 忽略普通的 Routing Table，直接将数据包送到远端主机上。
- R: 记录路由过程。
- s: 设置数据包的大小。
- t: 设置存活数值 TTL 的大小。
- v: 详细显示指令的执行过程。

ping 命令是使用最多的网络指令，通常我们使用它检测网络是否连通，它使用 ICMP 协议。但是有时会有这样的情况，我们可以浏览器查看一个网页，但是却无法 ping 通，这是因为一些网站处于安全考虑安装了防火墙。另外，也可以在自己计算机上试一试，通过下面的方法使系统对 ping 没有反应：

```
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

netstat

1. 作用

检查整个 Linux 网络状态。

2. 格式

netstat [-acCeFghilMnNoprstuvVwx][[-A][--ip]

3.主要参数

- a--all: 显示所有连线中的 Socket。
- A: 列出该网络类型连线中的 IP 相关地址和网络类型。
- c--continuous: 持续列出网络状态。
- C--cache: 显示路由器配置的快取信息。
- e--extend: 显示网络其它相关信息。
- F--fib: 显示 FIB。
- g--groups: 显示多重广播功能群组组员名单。
- h--help: 在线帮助。
- i--interfaces: 显示网络界面信息表单。
- l--listening: 显示监控中的服务器的 Socket。
- M--masquerade: 显示伪装的网络连线。
- n--numeric: 直接使用 IP 地址, 而不通过域名服务器。
- N--netlink--symbolic: 显示网络硬件外围设备的符号连接名称。
- o--timers: 显示计时器。
- p--programs: 显示正在使用 Socket 的程序识别码和程序名称。
- r--route: 显示 Routing Table。
- s--statistic: 显示网络工作信息统计表。
- t--tcp: 显示 TCP 传输协议的连线状况。
- u--udp: 显示 UDP 传输协议的连线状况。
- v--verbose: 显示指令执行过程。
- V--version: 显示版本信息。
- w--raw: 显示 RAW 传输协议的连线状况。
- x--unix: 和指定“-A unix”参数相同。
- ip--inet: 和指定“-A inet”参数相同。

4.应用实例

netstat 主要用于 Linux 察看自身的网络状况, 如开启的端口、在为哪些用户服务, 以及服务的状态等。此外, 它还显示系统路由表、网络接口状态等。可以说, 它是一个综合性的网络状态的察看工具。在默认情况下, netstat 只显示已建立连接的端口。如果要显示处于监听状态的所有端口, 使用-a 参数即可:

```
#netstat -a
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 *:32768 *:.* LISTEN
tcp 0 0 *:32769 *:.* LISTEN
tcp 0 0 *:nfs *:.* LISTEN
tcp 0 0 *:32770 *:.* LISTEN
tcp 0 0 *:868 *:.* LISTEN
tcp 0 0 *:617 *:.* LISTEN
tcp 0 0 *:mysql *:.* LISTEN
tcp 0 0 *:netbios-ssn *:.* LISTEN
tcp 0 0 *:sunrpc *:.* LISTEN
```

```
tcp 0 0 *:10000 *: LISTEN
```

```
tcp 0 0 *:http *: LISTEN
```

.....

上面显示出，这台主机同时提供 HTTP、FTP、NFS、MySQL 等服务。

telnet

1. 作用

telnet 表示开启终端机阶段作业，并登入远端主机。**telnet** 是一个 Linux 命令，同时也是一个协议（远程登陆协议）。

2. 格式

telnet [-8acdEfFKLrx][-b][-e][-k][-l][-n][-S][-X][主机名称 IP 地址]

3. 主要参数

- 8: 允许使用 8 位字符资料，包括输入与输出。
- a: 尝试自动登入远端系统。
- b: 使用别名指定远端主机名称。
- c: 不读取用户专属目录里的 **.telnetrc** 文件。
- d: 启动排错模式。
- e: 设置脱离字符。
- E: 滤除脱离字符。
- f: 此参数的效果和指定“-F”参数相同。
- F: 使用 Kerberos V5 认证时，加上此参数可把本地主机的认证数据上传到远端主机。
- k: 使用 Kerberos 认证时，加上此参数让远端主机采用指定的领域名，而非该主机的域名。
- K: 不自动登入远端主机。
- l: 指定要登入远端主机的用户名称。
- L: 允许输出 8 位字符资料。
- n: 指定文件记录相关信息。
- r: 使用类似 **rlogin** 指令的用户界面。
- S: 服务类型，设置 **telnet** 连线所需的 IP TOS 信息。
- x: 假设主机有支持数据加密的功能，就使用它。
- X: 关闭指定的认证形态。

4. 应用说明

用户使用 **telnet** 命令可以进行远程登录，并在远程计算机之间进行通信。用户通过网络在远程计算机上登录，就像登录到本地机上执行命令一样。为了通过 **telnet** 登录到远程计算机上，必须知道远程机上的合法用户名和口令。虽然有些系统确实为远程用户提供登录功能，但出于对安全的考虑，要限制来宾的操作权限，因此，这种情况下能使用的功能是很少的。

telnet 只为普通终端提供终端仿真，而不支持 X-Window 等图形环境。当允许远程用户登录时，系统通常把这些用户放在一个受限制的 Shell 中，以防系统被怀有恶意的或不小心的用户破坏。用户还可以使用 **telnet** 从远程站点登录到自己的计算机上，检查电子邮件、编辑文件和运行程序，就像在本地登录一样。

route

1. 作用
route 表示手工产生、修改和查看路由表。
2. 格式
#route [-add][-net|-host] targetaddress [-netmask Nm][dev]If
#route [-delete][-net|-host] targetaddress [gw Gw] [-netmask Nm] [dev]If
3. 主要参数
-add: 增加路由。
-delete: 删除路由。
-net: 路由到达的是一个网络，而不是一台主机。
-host: 路由到达的是一台主机。
-netmask Nm: 指定路由的子网掩码。
gw: 指定路由的网关。
[dev]If: 强迫路由链指定接口。
3. 应用实例
route 命令是用来查看和设置 Linux 系统的路由信息，以实现与其它网络的通信。要实现两个不同的子网之间的通信，需要一台连接两个网络的路由器，或者同时位于两个网络的网关来实现。
在 Linux 系统中，设置路由通常是为了解决以下问题：该 Linux 系统在一个局域网中，局域网中有一个网关，能够让机器访问 Internet，那么就需要将这台机器的 IP 地址设置为 Linux 机器的默认路由。使用下面命令可以增加一个默认路由：
route add 0.0.0.0 192.168.1.1

六、 Android 系统的超级终端命令大全（六）--系统安全相关命令

su

1. 作用
su 的作用是变更为其它使用者的身份，超级用户除外，需要键入该使用者的密码。
2. 格式
su [选项]... [-] [USER [ARG]...]
3. 主要参数
-f , --fast: 不必读启动文件（如 csh.cshrc 等），仅用于 csh 或 tcsh 两种 Shell。
-l , --login: 加了这个参数之后，就好像是重新登陆为该使用者一样，大部分环境变量（例如 HOME、SHELL 和 USER 等）都是以该使用者（USER）为主，并且工作目录也会改变。如果没有指定 USER，缺省情况是 root。
-m , -p , --preserve-environment: 执行 su 时不改变环境变数。
-c command: 变更账号为 USER 的使用者，并执行指令（command）后再变回原来使用者。
USER: 欲变更的使用者账号，ARG 传入新的 Shell 参数。
4. 应用实例
变更账号为超级用户，并在执行 df 命令后还原使用者。

su -c df root

umask

1. 作用

umask 设置用户文件和目录的文件创建缺省屏蔽值，若将此命令放入 profile 文件，就可控制该用户后续所建文件的存取许可。它告诉系统在创建文件时不给谁存取许可。使用权限是所有用户。

2. 格式

umask [-p] [-S] [mode]

3. 参数

—S: 确定当前的 umask 设置。

—p: 修改 umask 设置。

[mode]: 修改数值。

4. 说明

传统 Unix 的 umask 值是 022，这样就可以防止同属于该组的其它用户及别的组的用户修改该用户的文件。既然每个用户都拥有并属于一个自己的私有组，那么这种“组保护模式”就不在需要了。严密的权限设定构成了 Linux 安全的基础，在权限上犯错误是致命的。需要注意的是，umask 命令用来设置进程所创建的文件读写权限，最保险的值是 0077，即关闭创建文件的进程以外的所有进程的读写权限，表示为 -rw-----。在 ~/.bash_profile 中，加上一行命令 umask 0077 可以保证每次启动 Shell 后，进程的 umask 权限都可以被正确设定。

5. 应用实例

umask -S

u=rwx,g=rx,o=rx

umask -p 177

umask -S

u=rw,g=,o=

上述 5 行命令，首先显示当前状态，然后把 umask 值改为 177，结果只有文件所有者具有读写文件的权限，其它用户不能访问该文件。这显然是一种非常安全的设置。

chgrp

1. 作用

chgrp 表示修改一个或多个文件或目录所属的组。使用权限是超级用户。

2. 格式

chgrp [选项]... 组 文件...

或

chgrp [选项]... --reference=参考文件 文件...

将每个的所属组设定为。

3. 参数

-c, --changes : 像 --verbose，但只在有更改时才显示结果。

--dereference: 会影响符号链接所指示的对象，而非符号链接本身。

-h, --no-dereference: 会影响符号链接本身, 而非符号链接所指示的目的地(当系统支持更改符号链接的所有者, 此选项才有效)。

-f, --silent, --quiet: 去除大部分的错误信息。

--reference=参考文件: 使用的所属组, 而非指定的。

-R, --recursive: 递归处理所有的文件及子目录。

-v, --verbose: 处理任何文件都会显示信息。

4.应用说明

该命令改变指定指定文件所属的用户组。其中 **group** 可以是用户组 ID, 也可以是/etc/group 文件中用户组的组名。文件名是以空格分开的要改变属组的文件列表, 支持通配符。如果用户不是该文件的属主或超级用户, 则不能改变该文件的组。

6. 应用实例

改变/opt/local /book/及其子目录下的所有文件的属组为 **book**, 命令如下:

```
$ chgrp -R book /opt/local /book
```

chmod

1. 作用

chmod 命令是非常重要的, 用于改变文件或目录的访问权限, 用户可以用它控制文件或目录的访问权限, 使用权限是超级用户。

2. 格式

chmod 命令有两种用法。一种是包含字母和操作符表达式的字符设定法(相对权限设定); 另一种是包含数字的数字设定法(绝对权限设定)。

(1) 字符设定法

chmod [who] [+|-|=] [mode] 文件名

◆ 操作对象 **who** 可以是下述字母中的任一个或它们的组合

u: 表示用户, 即文件或目录的所有者。

g: 表示同组用户, 即与文件属主有相同组 ID 的所有用户。

o: 表示其它用户。

a: 表示所有用户, 它是系统默认值。

◆ 操作符号

+: 添加某个权限。

-: 取消某个权限。

=: 赋予给定权限, 并取消其它所有权限(如果有的话)。

◆ 设置 **mode** 的权限可用下述字母的任意组合

r: 可读。

w: 可写。

x: 可执行。

X: 只有目标文件对某些用户是可执行的或该目标文件是目录时才追加 **x** 属性。

s: 文件执行时把进程的属主或组 ID 置为该文件的文件属主。方式“**u+s**”设置文件的用户 ID 位, “**g+s**”设置组 ID 位。

t: 保存程序的文本到交换设备上。

u: 与文件属主拥有一样的权限。

g: 与文件属主同组的用户拥有一样的权限。

o: 与其它用户拥有一样的权限。

文件名: 以空格分开的要改变权限的文件列表, 支持通配符。

一个命令行中可以给出多个权限方式, 其间用逗号隔开。

(2) 数字设定法

数字设定法的一般形式为:

chmod [mode] 文件名

数字属性的格式应为 3 个 0 到 7 的八进制数, 其顺序是(u)(g)(o)文件名, 以空格分开的要改变权限的文件列表, 支持通配符。

数字表示的权限的含义如下: 0001 为所有者的执行权限; 0002 为所有者的写权限; 0004 为所有者的读权限; 0010 为组的执行权限; 0020 为组的写权限; 0040 为组的读权限; 0100 为其他人的执行权限; 0200 为其他人的写权限; 0400 为其他人的读权限; 1000 为粘贴位置位; 2000 表示假如这个文件是可执行文件, 则为组 ID 为位置位, 否则其中文件锁定位置位; 4000 表示假如这个文件是可执行文件, 则为用户 ID 为位置位。

3. 实例

如果一个系统管理员写了一个表格(tem)让所有用户填写, 那么必须授权用户对这个文件有读写权限, 可以使用命令: `# chmod 666 tem`

上面代码中, 这个 666 数字是如何计算出来的呢? 0002 为所有者的写权限, 0004 为所有者的读权限, 0020 为组的写权限, 0040 为组的读权限, 0200 为其他人的写权限, 0400 为其他人的读权限, 这 6 个数字相加就是 666 (注以上数字都是八进制数)。

如果用字符权限设定使用下面命令:

`#chmod a=wx tem`

chown

1. 作用

更改一个或多个文件或目录的属主和属组。使用权限是超级用户。

2. 格式

chown [选项] 用户或组 文件

3. 主要参数

--dereference: 受影响的是符号链接所指示的对象, 而非符号链接本身。

-h, --no-dereference: 会影响符号链接本身, 而非符号链接所指示的目的地(当系统支持更改符号链接的所有者, 此选项才有效)。

--from=目前所有者:目前组只当每个文件的所有者和组符合选项所指定的, 才会更改所有者和组。其中一个可以省略, 这已省略的属性就不需要符合原有的属性。

-f, --silent, --quiet: 去除大部分的错误信息。

-R, --recursive: 递归处理所有的文件及子目录。

-v, --verbose: 处理任何文件都会显示信息。

4. 说明

chown 将指定文件的拥有者改为指定的用户或组, 用户可以是用户名或用户 ID; 组可以是组名或组 ID; 文件是以空格分开的要改变权限的文件列表, 支持

通配符。系统管理员经常使用 **chown** 命令，在将文件拷贝到另一个用户的目录下以后，让用户拥有使用该文件的权限。

5.应用实例

- 1) 把文件 **shiyang.c** 的所有者改为 **wan**

```
$ chown wan shiyang.c
```

- 2) 把目录 **/hi** 及其下的所有文件和子目录的属主改成 **wan**，属组改成 **users**。

```
$ chown -R wan.users /hi
```

chattr

1. 作用

修改 **ext2** 和 **ext3** 文件系统属性(attribute)，使用权限超级用户。

2. 格式

chattr [-RV] [-+=AacDdijsSu] [-v version] 文件或目录

3. 主要参数

—**R**: 递归处理所有的文件及子目录。

—**V**: 详细显示修改内容，并打印输出。

—: 失效属性。

+: 激活属性。

= : 指定属性。

A: **Atime**，告诉系统不要修改对这个文件的最后访问时间。

S: **Sync**，一旦应用程序对这个文件执行了写操作，使系统立刻把修改的结果写到磁盘。

a: **Append Only**，系统只允许在这个文件之后追加数据，不允许任何进程覆盖或截断这个文件。如果目录具有这个属性，系统将只允许在这个目录下建立和修改文件，而不允许删除任何文件。

i: **Immutable**，系统不允许对这个文件进行任何的修改。如果目录具有这个属性，那么任何的进程只能修改目录之下的文件，不允许建立和删除文件。

D: 检查压缩文件中的错误。

d: **No dump**，在进行文件系统备份时，**dump** 程序将忽略这个文件。

C: **Compress**，系统以透明的方式压缩这个文件。从这个文件读取时，返回的是解压之后的数据；而向这个文件中写入数据时，数据首先被压缩之后才写入磁盘。

s: **Secure Delete**，让系统在删除这个文件时，使用 0 填充文件所在的区域。

u: **Undelete**，当一个应用程序请求删除这个文件，系统会保留其数据块以便以后能够恢复删除这个文件。

4. 说明

chattr 命令的作用很大，其中一些功能是由 **Linux** 内核版本来支持的，如果 **Linux** 内核版本低于 2.2，那么许多功能不能实现。同样—**D** 检查压缩文件中的错误的功能，需要 2.5.19 以上内核才能支持。另外，通过 **chattr** 命令修改属性能够提高系统的安全性，但是它并不适合所有的目录。**chattr** 命令不能保护 **/**、**/dev**、**/tmp**、**/var** 目录。

5.应用实例

- 1) 恢复 **/root** 目录,即子目录的所有文件

```
# chattr -R +u/root
```

2) 用 **chattr** 命令防止系统中某个关键文件被修改

在 Linux 下, 有些配置文件(**passwd**, **fstab**)是不允许任何人修改的, 为了防止被误删除或修改, 可以设定该文件的“不可修改位(**immutable**)”, 命令如下:

```
# chattr +i /etc/fstab
```

ps

1. 作用

ps 显示瞬间进程 (**process**) 的动态, 使用权限是所有使用者。

2. 格式

ps [options] [--help]

3. 主要参数

ps 的参数非常多, 此出仅列出几个常用的参数。

-A: 列出所有的进程。

-l: 显示长列表。

-m: 显示内存信息。

-w: 显示加宽可以显示较多的信息。

-e: 显示所有进程。

a: 显示终端上的所有进程, 包括其它用户的进程。

-au: 显示较详细的信息。

-aux: 显示所有包含其它使用者的进程。

4. 说明

要对进程进行监测和控制, 首先要了解当前进程的情况, 也就是需要查看当前进程。**ps** 命令就是最基本、也是非常强大的进程查看命令。使用该命令可以确定有哪些进程正在运行、运行的状态、进程是否结束、进程有没有僵尸、哪些进程占用了过多的资源等。

4. 应用实例

在进行系统维护时, 经常会出现内存使用量惊人, 而又不知道是哪一个进程占用了大量进程的情况。除了可以使用 **top** 命令查看内存使用情况之外, 还可以使用下面的命令:

```
ps aux | sort +5n
```

七、 其他命令

tar

1. 作用

tar 命令是 Unix/Linux 系统中备份文件的可靠方法, 几乎可以工作于任何环境中, 它的使用权限是所有用户。

2. 格式

tar [主选项+辅选项] 文件或目录

3. 主要参数

使用该命令时，主选项是必须要有的，它告诉 **tar** 要做什么事情，辅选项是辅助使用的，可以选用。

主选项：

-c 创建新的档案文件。如果用户想备份一个目录或是一些文件，就要选择这个选项。

-r 把要存档的文件追加到档案文件的末尾。例如用户已经做好备份文件，又发现还有一个目录或是一些文件忘记备份了，这时可以使用该选项，将忘记的目录或文件追加到备份文件中。

-t 列出档案文件的内容，查看已经备份了哪些文件。

-u 更新文件。就是说，用新增的文件取代原备份文件，如果在备份文件中找不到要更新的文件，则把它追加到备份文件的最后。

-x 从档案文件中释放文件。

辅助选项：

-b 该选项是为磁带机设定的，其后跟一数字，用来说明区块的大小，系统预设值为 20 (20×512 bytes)。

-f 使用档案文件或设备，这个选项通常是必选的。

-k 保存已经存在的文件。例如把某个文件还原，在还原的过程中遇到相同的文件，不会进行覆盖。

-m 在还原文件时，把所有文件的修改时间设定为现在。

-M 创建多卷的档案文件，以便在几个磁盘中存放。

-v 详细报告 **tar** 处理的文件信息。如无此选项，**tar** 不报告文件信息。

-w 每一步都要求确认。

-z 用 **gzip** 来压缩/解压缩文件，加上该选项后可以将档案文件进行压缩，但还原时也一定要使用该选项进行解压缩。

4.应用说明

tar 是 Tape Archive (磁带归档) 的缩写，最初设计用于将文件打包到磁带上。如果下载过 Linux 的源代码，或许已经碰到过 **tar** 文件请注意，不要忘了 Linux 是区分大小写的。例如，**tar** 命令应该总是以小写的形式执行。命令行开关可以是大写、小写或大小写的混合。例如，-t 和-T 执行不同的功能。文件或目录名称可以混合使用大小写，而且就像命令和命令行开关一样是区分大小写的。

4. 应用实例

tar 是一个命令行的工具，没有图形界面。使用 **Konsole** 打开一个终端窗口，接下来是一个简单的备份命令（在 /temp 目录中创建一个 back.tar 的文件，/usr 目录中所有内容都包含在其中。）：

```
$tar cvf - /usr > /temp/back.tar
```

另外，**tar** 命令支持前面第三讲中讲过的 **crontab** 命令，可以用 **crontab** 工具设置成基于时间的有规律地运行。例如，每晚 6 点把 /usr 目录备份到 hda—第一个 IDE 接口的主驱动器（总是位于第一个硬盘）中，只要将下面语句添加到 root 的 **crontab** 中即可：

```
$00 06 * * * tar cvf /dev/hda1/usrfiles.tar - /usr
```

一般情况下，以下这些目录是需要备份的：

◆ /etc 包含所有核心配置文件，其中包括网络配置、系统名称、防火墙规则、用户、组，以及其它全局系统项。

◆ /var 包含系统守护进程（服务）所使用的信息，包括 DNS 配置、DHCP 租期、邮件缓冲文件、HTTP 服务器文件、dB2 实例配置等。

◆ /home 包含所有默认用户的主目录，包括个人设置、已下载的文件和用户不希望失去的其它信息。

◆ /root 根（root）用户的主目录。

◆ /opt 是安装许多非系统文件的地方。IBM 软件就安装在这里。OpenOffice、JDK 和其它软件在默认情况下也安装在这里。

有些目录是可以不备份的：

◆ /proc 应该永远不要备份这个目录。它不是一个真实的文件系统，而是运行内核和环境的虚拟化视图，包括诸如/proc/kcore 这样的文件，这个文件是整个运行内存的虚拟视图。备份这些文件只是在浪费资源。

◆ /dev 包含硬件设备的文件表示。如果计划还原到一个空白的系统，就可以备份/dev。然而，如果计划还原到一个已安装的 Linux 系统，那么备份/dev 是没有必要的。

unzip

1. 作用

unzip 命令位于/usr/bin 目录中，它们和 MS DOS 下的 pkzip、pkunzip 及 MS Windows 中的 Winzip 软件功能一样，将文件压缩成.zip 文件，以节省硬盘空间，当需要的时候再将压缩文件用 unzip 命令解开。该命令使用权限是所有用户。

2. 格式

unzip [-cflptuvz][-agCjLMnoqsVX][-P][.zip 文件][文件][-d][-x]

3. 主要参数

- c: 将解压缩的结果显示到屏幕上，并对字符做适当的转换。
- f: 更新现有的文件。
- l: 显示压缩文件内所包含的文件。
- p: 与-c 参数类似，会将解压缩的结果显示到屏幕上，但不会执行任何的转换。
- t: 检查压缩文件是否正确。
- u: 与-f 参数类似，但是除了更新现有的文件外，也会将压缩文件中的其它文件解压缩到目录中。
- v: 执行是时显示详细的信息。
- z: 仅显示压缩文件的备注文字。
- a: 对文本文件进行必要的字符转换。
- b: 不要对文本文件进行字符转换。
- C: 压缩文件中的文件名称区分大小写。
- j: 不处理压缩文件中原有的目录路径。
- L: 将压缩文件中的全部文件名改为小写。
- M: 将输出结果送到 more 程序处理。
- n: 解压缩时不要覆盖原有的文件。
- o: 不必先询问用户，unzip 执行后覆盖原有文件。
- P: 使用 zip 的密码选项。
- q: 执行时不显示任何信息。

-s: 将文件名中的空白字符转换为底线字符。
-V: 保留 VMS 的文件版本信息。
-X: 解压缩时同时回存文件原来的 UID/GID。
[.zip 文件]: 指定.zip 压缩文件。
[文件]: 指定要处理.zip 压缩文件中的哪些文件。
-d: 指定文件解压缩后所要存储的目录。
-x: 指定不要处理.zip 压缩文件中的哪些文件。
-Z unzip: -Z 等于执行 zipinfo 指令。在 Linux 中, 还提供了一个叫 zipinfo 的工具, 能够察看 zip 压缩文件的详细信息。

gunzip

1. 作用

gunzip 命令作用是解压文件, 使用权限是所有用户。

2. 格式

gunzip [-acfhLlNqrtvV][-s][文件...]

或者

gunzip [-acfhLlNqrtvV][-s][目录]

3. 主要参数

-a 或--ascii: 使用 ASCII 文字模式。
-c 或--stdout 或--to-stdout: 把解压后的文件输出到标准输出设备。
-f 或-force: 强行解开压缩文件, 不理睬文件名称或硬连接是否存在, 以及该文件是否为符号连接。
-h 或--help: 在线帮助。
-l 或--list: 列出压缩文件的相关信息。
-L 或--license: 显示版本与版权信息。
-n 或--no-name: 解压缩时, 若压缩文件内含有原来的文件名称及时间戳记, 则将其忽略不予处理。
-N 或--name: 解压缩时, 若压缩文件内含有原来的文件名称及时间戳记, 则将其回存到解开的文件上。
-q 或--quiet: 不显示警告信息。
-r 或--recursive: 递归处理, 将指定目录下的所有文件及子目录一并处理。
-S 或--suffix: 更改压缩字尾字符串。
-t 或--test: 测试压缩文件是否正确无误。
-v 或--verbose: 显示指令执行过程。
-V 或--version: 显示版本信息。

4. 说明

gunzip 是个使用广泛的解压缩程序, 它用于解开被 gzip 压缩过的文件, 这些压缩文件预设最后的扩展名为 “.gz”。事实上, gunzip 就是 gzip 的硬连接, 因此不论是压缩或解压缩, 都可通过 gzip 指令单独完成。gunzip 最新版本是 1.3.3。